

"Keys to a Successful Services Engagement"

Speaker:

Don Harbin

Director, Professional Services


MontaVista Software

Moderator:

Curt Schwaderer, OpenSystems Media



Agenda



Quick tour of viewer tools
A few topical thoughts
Speaker Presentation
Q&A Session

Development Challenges

- Increased TTM Pressures
- New Technologies
- Short-Staffed





Keys to a Successful Services Engagement

Don Harbin
Director, Professional Services

- **Overview of Professional Services for embedded Linux**
 - Common problems encountered in embedded Linux development
 - Leveraging the open source community in creating custom solutions
 - Knowing the community
 - Working in the community
 - Leveraging the community
 - Protecting one's assets: Understanding the GPL
- **MontaVista Professional Services Solutions Offerings**



- **General Definition:** A department of a software supplier that provides consultancy, resourcing, and expertise within a software domain in one or more of the following disciplines:
 - Software architecture
 - Software design
 - Programming
 - Integration
 - Validation & test
 - Quality
- **In this case, the domain expertise is embedded Linux. Additional services that are of value in this domain include:**
 - Licensing / Legal
 - Linux community navigation
 - Community Projects, Stable Versions, etc.



- **Starting Engagements Late**

- Customers come when have tried on their own and failed. Then no time left!
- Waiting too long creates the scenario wherein
 - Up front planning and requirements are sacrificed.
 - Quality is sacrificed

- **Farming out one's Intellectual Property**

- Engage instead in a service that farms out all else so that team can focus on company assets

- **Chasing the Kernel!**

- Select, Stop and Stabilize on a Kernel
 - Choose the right version and stabilize!

- **Not Embracing the Community**

- Processes, standards, and projects
- Attempting to force Linux to conform to internal processes can waste many cycles
- Code Control System, Coding Standards, Peer Review methods, Check-in processes, etc.
- Leverage the community: push fixes into the community, get fixes from the community

- **Jumping too quickly into development**

- Understand and develop the solution up-front
- Consider the leverage that the Community can provide
 - Existing projects in the Community are key

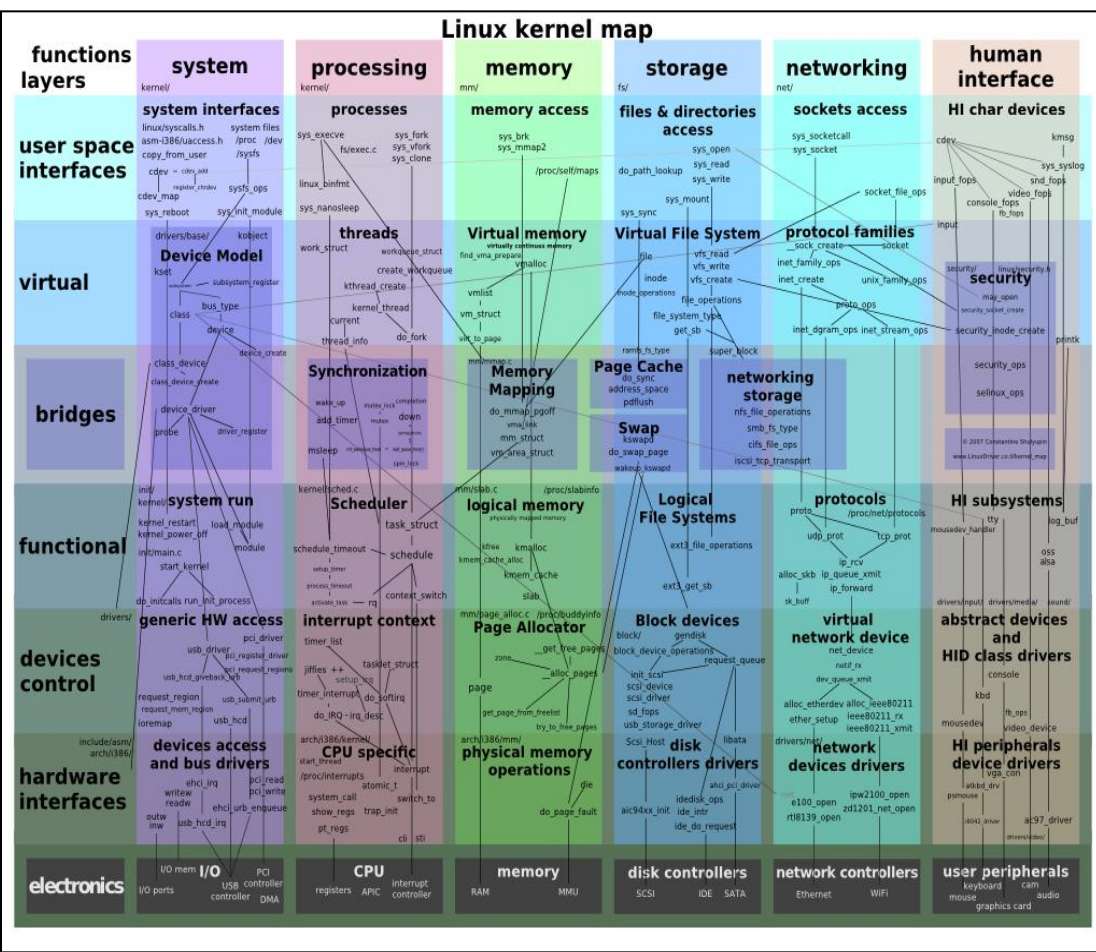


- **Evaluate tradeoffs of offshoring solutions and maintaining the appropriate blend**

- Local presence important in some, but not all engagements



Knowing the Community



- **Wikipedia definition:** The Linux kernel is an operating system kernel used by the Linux family of Unix-like operating systems. The term Linux distribution is used to refer to the various operating systems that run on top of the Linux kernel.
- The later kernels consist of over 11M lines of code contributed by nearly 1,200 engineers
- The kernel typically releases a new version every 10 weeks
- Less than 5% of the contributions to Linux are targeted for the embedded market
- There are hundreds of projects in open source that leverage the kernel but do not reside within it

11M lines of code, rev'd every 10 weeks by ~1200 engineers, less than 5% aligned to embedded, with hundreds of projects influencing/leveraging the results

1.1 CORE UTILITIES

Anacron	apt-rpm	at	audit	base-file-DEV
Base-passwd	bash	bc	bootpc	busybox
bzip2	console-data	console-tools	coreutils	cpio
cracklib	cron	dash	dbus	dhcpcd
diff	dnrnd	dump	ed	file
findutils	gawk	grep	groff	gzip
hotplug	ifupdown	ilmtk	lcap	less
Libcap	libpcap	linuxinfo	logrotate	lrzsz
Isb-helpers	Isdf	madev	man-db	man-pages
memstat	mgetty	minicom	module-init-tools	
mvluits	nano	nvi	pam	pax
pidentd	portmap	procps	psmisc	rpm
rpm-edition	schedutils	sed	setserial	
shadow	socket	sudo	sysfsutils	
syslogd	sysutils	sysvinit		
tar	tcp-wrappers			

1.2 CORE NETWORKING

ethtool	ipgrab	iproute	ipsec-tools
iptables	iputils	mailx	
mdnsresponder	netbase	net-base	
netkit-ftp	netkit-telnet	net-ftp	
ntp	openssh	ppp	
rp-pppoe	rstatd	rsync	
tcpdump	ftpp-hpp	ftplib	

1.2 SECURITY

libselinux	libsemanage	libsepol	lidstools
selinux-policy			

2.1 KERNEL FEATURES - SOUND

alsa-utils	aumix	libasound	oss-fx
------------	-------	-----------	--------

2.2 KERNEL FEATURES - BLUETOOTH

bluez-hcidump	bluez-libs	bluez-utils
---------------	------------	-------------

2.3 KERNEL FEATURES - FLASH FILESYSTEMS

mt-d-utils	squashfs-tools	yaffs-utils
------------	----------------	-------------

2.4 KERNEL FEATURES - GENERAL FILESYSTEMS

autofs	dosfstools	e2fsprogs	ecryptfs-utils	jfsutils
libattr	libdmapi	nfs-utils	xfsdump	xfsprogs

2.5 KERNEL FEATURES - WIRELESS NETWORKING

hostapd	hostap-utils	kismet	wireless-tools
---------	--------------	--------	----------------

2.6 KERNEL FEATURES - MISC

cyclctest	eject	hdparm	libusb	lksctp-rioutils
toolsopen-iscsi	parted	pciutils	pcmciautils	
usbutils	vlan	watchdog		

3.1 DEVELOPMENT - LIBRARIES

beecrypt	db	editline-libs	elfutils	expat
Flex	gd	gdbm	gettext	glib
gnutls	gpp	libcurl	libevent	libgrypt
libpgg-error	libIDL-libs	libblockfile	libpaper	libpcre
libtool	libwww	libxml2	ncurses	neon
openssl	ORBit2	preempt	readline-libs	sqlite

3.2 DEVELOPMENT - SCRIPT LANGUAGES

expect	perl	php4	python	tcl
Tcsh				

3.3 DEVELOPMENT - RUNTIME DEVELOPMENT TOOLS

autoconf	automake	bison	checkpolicy	dtc
M4	make	flex	p2linux	patch
	pkgconfig	policycoreutils	texinfo	v2linux
	gettext	ltd-control		

3.4 DEVELOPMENT - TEST/DEBUG TOOLS

degnu	dmalloc	doxygen
gdb	ksymoops	libunwind
mpatrol	nana	oprofile
prelink	strace	

4.1 SERVICES AND ADDITIONAL UTILITIES

adduser	apache	atm
bind	bonnie++	bridge-utils
bsd-finger	busybox-static	cproto
cups	dhcp	dialog
gpm	jpreempt	lmbench
lynx	netkit-ntalk	netkit-routed
net-snmp	nis	nss_idap
openobex	postfix	postgresql
procmail	quota	radvd
rdist	samba	sharutils
sysstat	thttpd	udev
xmlsec	Freetype	libpn
fbset	fontconfig	pango
libjpeg	libmng	xorg-
libungif	matchbox	netperf
utempter	xbdb	openldap

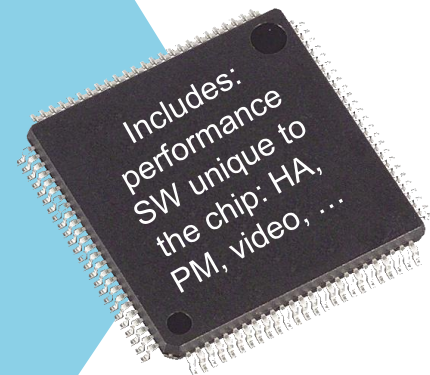
STANDARDS COMPLIANCE

CELF	CGL	OpenSAF	SAForum	OpenAIS
OpenHPI	PICMIG	LSB	MLI	Limo
SD	Plum Hall	Oracle		

There are hundreds of open source projects from which to leverage! Totaling over 40M LOC



1.1 CORE UTILITIES Anacron apt-rpm at audit base-file-DEV Base-passwd bash bc bootpc busybox bzip2 console-data console-tools coreutils cracklib cron dash dbus dhcpcd diff dndr dump ed file findutils gawk grep groff hotplug ifupdown ilmtk lcap Libcap libpcap linuxinfo logrotate lsb-helpers lsof makedev man-db memstat mgetty minicom module-init-tools mvlutils nano nvi pam pax pident portmap procps psmisc rpm rpm-edition schedutils sed setserial shadow socket sudo sysfsutils syslogd sysutils sysvinit tar tcp-wrappers				3.1 DEVELOPMENT - LIBRARIES beecrypt db editline-libs elfutils expat Flex gd gdbm gettext glib gnutils gpp libcurl libevent libgcript libgpg-error libIDL-libs libblockfile libpaper libpcrc libtool libwww libxml2 ncurses neon openssl ORBit2 preempt readline-libs sqlite 3.2 DEVELOPMENT - SCRIPT LANGUAGES expect perl php4 python tcl Tcsh			
1.2 CORE NETWORKING ethtool ipgrab iproute ipsec-tools iptables iputils mailx mdnsresponder netbase netkit-base netkit-ftp netkit-rsh netkit-telnet net-tools ntp openssh ppp rdate rp-pppoe rstatd rsync tcng tcpdump tftpd-hpa ip-tables				3.3 DEVELOPMENT - RUNTIME DEVELOPMENT TOOLS autoconf automake bison checkpolicy dtc M4 make nasmm p2linux patch pkgconfig policycoreutils texinfo v2linux genevent ltt-control			
1.2 SECURITY libselinux libsemanage libsepol lid3tools				3.4 DEV - RUNTIME TEST/DEBUG TOOLS dejagnu dmalloc doxygen gdb ksymoops libunwind mpatrol nana oprofile prelink strace			
2.1 KERNEL FEATURES - SOUND alsa-utils aumix libasound libaudiofile				4.1 SERVERS AND ADDITIONAL UTILITIES adduser apache atm bind bonnie++ bridge-utils bsd-finger busybox-static cproto cups dhcp dialog gpm jpreempt lmbench lynx netkit-ntalk netkit-routed net-snmp nis nss_idap openobex postfix postgresql procmail rdist samba sharutils sysstat xmisc thttpd udev x11libpng xinetd Freetype libpng cairo fbgetty fbset fontconfig gtk2 icewm libjpeg libmng x11libpng libtiff libungif libmng rxvt tslib utempter xkbd			
2.2 KERNEL FEATURES - BLUETOOTH bluez-hcidump bluez-libs bluez-utils				STANDARDS COMPLIANCE CELF CGL OpenSAF SAForum OpenAIS OpenHPI PICMIG LSB MLI Limo SD Plum Hall Oracle			
2.3 KERNEL FEATURES - FLASH FILESYSTEMS mtd-utils squashfs-tools yaffs-utils				2.4 KERNEL FEATURES - GENERAL FILESYSTEMS autofs dosfstools e2fsprogs ecryptfs-utils jfsutils libattr libdmapi nfs-utils xfsdump xfsprogs			
2.5 KERNEL FEATURES - WIRELESS NETWORKING hostapd hostapd-utils kismet wireless-tools				2.6 KERNEL FEATURES - MISC cyclictst eject hdparm libusb lksctp-rioutils toolsopen-iscsi parted pciutils pcmciautils usbutils vlan watchdog			
Arm		X86		PowerPC		MIPS	



Embedded Linux Services much consider

- Performance
- Portability
- Supportability



Knowing the Community

Working in and Leveraging the Community

- **Working in the Community**

- Community coding standards
- How to successfully contribute to projects



- **Leveraging the Community**

- Leveraging the community for defect resolution
- Maturity of varying projects



Protecting One's Assets

Understanding the GPL

- **Disclaimer #1: I am not a lawyer**
 - I am an engineer
 - Every engineer in the Linux domain think we understand the GPL and are self-appointed lawyers! 😊
- **But in general, following certain disciplines are great guidelines for Linux development**
- **All software developed / used should go through a Certificate Of Origin (COO) process to assure it can be legally used. Not just from GPL, but other licenses as well.**
- **Every company should have a legal position by which they stand on from legal counsel**

- **Kernel**

- Kernel Intellectual Property (IP) are commonly considered to be protected if using dynamically loaded kernel modules (DLKM)

- **Applications**

- Create new code!
- Make sure is any protected code of the OS copied into the application program
- Interface with the Linux kernel through the publicly exported APIs / using standard system calls



- **Overview of LKMs as derived (or not derived) works**
 - Argument for derived work: LKMs effectively become part of the running kernel when loaded.
 - It is essentially the same program whether you load it dynamically or statically
 - What the FSF would call “a distinction without a difference”
 - Argument against derived work: by interacting with the host solely through the public APIs, copyright law (and thus derivative work issues) may not even be applicable.
 - It’s closer to a user-space program, communicating with the kernel via standard system calls.
 - This **seems be accepted as a best practice** in the Linux industry.

- **Each Company must develop their own legal position on these, and other, cases**
 - Applications
 - Kernel Drivers
 - Plug-ins
 - GPL-only Flags
 - Libraries and the LGPL
 - Header Files
 - Code Creation Tools
 - Pipes
 - Sockets
 - What is and is not a derivative work?
 - When and how to publish GPL-developed code?





MontaVista PS Advantages

• MontaVista Leverage

- Existing Products
 - MVL6
 - DevRocket
- Advanced Validation & Test Facilities
 - Provides high quality results
- Provide Understanding the community
 - What tools are available that can help
 - What Open Source Components are available and useful
 - Stable and Mature?



• MontaVista Maturity

- Project development for over 10 years!
- Experienced Staff
- Diverse solutions
 - Multiple technologies
 - MIPS, PPC, ARM, x86
 - Multicore
 - Various S/W solutions
 - Kernel
 - Drivers
 - Middleware
 - Apps
 - High reliability
 - High Performance
 - Real-time
 - Various industries
 - Telecom
 - Mobile
 - Automotive
 - Android
 - Medical

- **MontaVista Staff**

- Diverse Staff
 - Skills
 - Timezones
 - Worldwide
- Competitive rates
 - Multiple development centers in competitive locations
- Many years of experience



Bringing Order to the Chaos

- **Compliments / Enhances MontaVista Tools**
 - Can, however, be independent
- **Embedded Linux Focus**
 - New or community code
 - Proprietary or Open Source
 - All processor technologies
 - Many software variants
- **Flexible contracts**
 - Fixed price or Time & Material
 - On-site or off-site
 - One developer or large-scale contracts
- **Out-source the common components allowing your team to focus on your core competencies**

- **Architecture and Requirements Definition**
- **Software Design**
- **Software Implementation**
- **Software Integration**
- **Software Validation & Test Services**
- **Productization**
 - Quality from Concept to Release
 - Quality throughout all phases of development
- **Toolchain Support**
- **Custom Tuning**
 - Performance enhancements
 - Code optimization
 - Real-Time Support
- **Post-Production Support**
 - Phone Support
 - Dedicated Support
 - On-Site Support
- **Open Source Project Development & Support**
 - Android
 - Networking
- **Expert Legal Advice in regards to Linux Licensing and solutions**
- **Skilled project planning and oversight**
 - Cost and schedule containment
- **Training**



Questions / Comments / Open Discussion

- **QA process and systems**

- June 21st
- Joe Pearson, Director of QA and Build -

- **Standards and the Automotive Infotainment Market**

- July 21st
- Dan Cauchy, VP Marketing and Business Development



Thank You



Don Harbin

Director, Professional Services

MontaVista Software

Q

&

A



Thanks for joining us



Event archive available at:

<http://ecast.opensystemsmedia.com/>

E-mail us at: clong@opensystemsmedia.com